

```
int sumto(int n)
{
    int i;
    int sum = 0;

    for (i = 0; i < n; ++i)
        sum += i;
    return sum;
}
```

Figure 1

Figure 1: sumto.c

```
int sumto(int n)
{
    int i;
    int sum = 0;

    for (i = 0; i < n; ++i)
        { sum += i; yield(); }
    return sum;
}
```

Figure 2

Figure 2: The sumto function

```

int sumto(int n)
{
    int i;
    int sum = 0;
    int tripct = 10;

    for (i = 0; i < n; ++i)
    {
        sum += i;
        if (0 == --tripct)
            { tripct = 10; yield(); }
    }
    return sum;
}

```

Figure 3

Figure 3: sumto.c

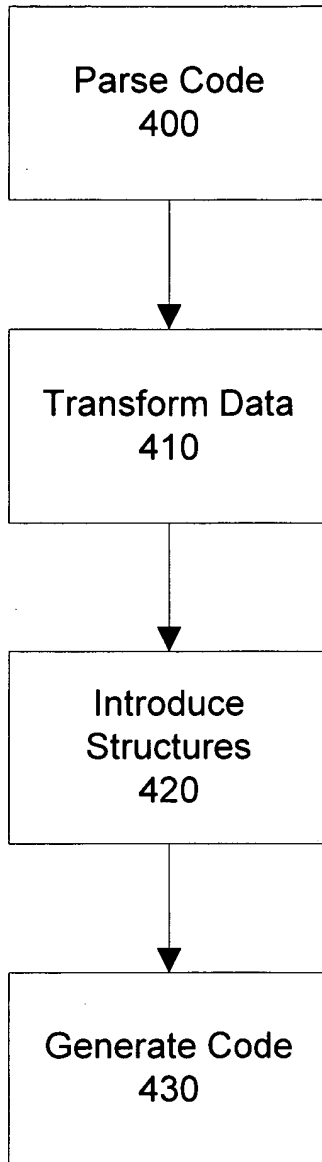


Figure 4

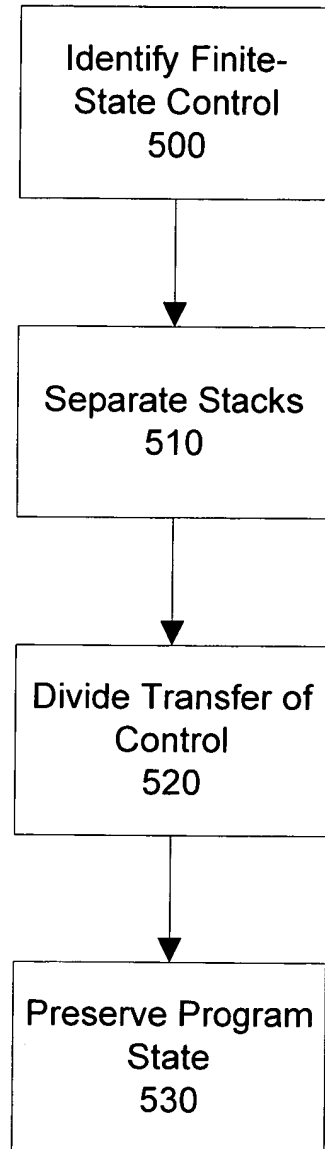


Figure 5

```

typedef struct context {
    int n; int i; int sum;
} context;

typedef struct Continuation {
    int state; context ctxt;
} Continuation;

bool sumto_cps(Continuation* k)
{
    int      state = k->state;
    context* ctxt  = &k->ctxt;

    switch (state) {
    case 0:  ctxt->sum = 0; ctxt->i = 0;
            k->state = 1;
            return false;
    case 1:  { bool tst = (ctxt->i < ctxt->n);
            if (tst) k->state = 2;
            else    k->state = 3; }
            return false;
    case 2:  ctxt->sum += ctxt->i; ++ctxt->i;
            k->state = 1;
            return false;
    case 3:  return true;
    }
}

int sumto(int n) {
    Continuation k;
    bool done = false;
    k.ctxt.n = n; k.state = 0;
    while (!done)
        done = sumto_cps(&k);
    return k.ctxt.sum;
}

```

Figure 6

```

WhatNextType
sumtoSLT (int caselab) {
    switch (caselab) {
        ...
        case 2:
            arg2 = POP(void*); arg1 = POP(void*);
            arg0 = POP(void*);
            kCaselab = POP(int);
            kFunc = POP(WhatNextType (*)(int ));
        case 3:
            idlab2:
            {
                int n = (int )(arg0);
                int i = (int )(arg1);
                int sum = (int )(arg2);
                if (i < n) {
                    sum = sum + i; ++i;
                    arg0 = (void *) (n);
                    arg1 = (void *) (i);
                    arg2 = (void *) (sum);
                    if (--ticks > 0) goto idlab2;
                    PUSH(kFunc); PUSH(kCaselab);
                    PUSH(arg0);
                    PUSH(arg1); PUSH(arg2);
                    PUSH(sumtoSLT); PUSH(2);
                    return STOP;
                } else {
                    arg0 = (void *) (sum);
                    if (--ticks > 0) goto idlab4;
                    PUSH(kFunc); PUSH(kCaselab);
                    PUSH(arg0);
                    PUSH(sumtoSLT); PUSH(4);
                    return STOP;
                }
            }
            break;
        ...
    }
}

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

Figure 7